Construction Rovers

Kendall Johnson

kjohns21@masonlive.gmu.edu || Youtube: MoveOverRover ||Instagram: Moveroverrover1
George Mason University | 4400 University Drive, Fairfax, VA 22030
Goddard Space Flight Lab | 8800 Greenbelt Rd, Greenbelt, MD 20771

Abstract

In July 2019, I attended the *Lunar Operations and Technologies to Enable Human Exploration of Mars and the Moon* Workshop. It is a group of NASA scientists who believe that leveraging lunar technologies will allow us to be more successful in getting to Mars. They also believe in creating devices that can work on both celestial bodies will save time, money, and energy in our inevitable journey into the cosmos. One of the main priorities stated for helping to enable Moon and Mars exploration is specified rovers that can complete tasks involving science, maintenance, transportation, and construction (Thronson, 2019). NASA's Martian rover team has done a great job concerning a scientific rover, the classic Lunar rover can transport humans on the surface of the Moon, and currently, there is nothing to maintain on any celestial body, other than Earth, for a maintenance rover. The last of this Lunar Operations and Technologies task is a construction rover, and this is where I have started my theories and experiments to demonstrate actions, hard data, possibilities, shortcomings, and abilities of a future construction rover by building and testing a prototype.

The first task of a construction rover is to best prepare the celestial body's surface for the arrival of humans. The rover is given the task of preparing the surface that can include clearing away regolith, flattening the ground, and moving equipment and debris. This is a demonstration of the proof of concept for the construction rover as an autonomous platform using Artificial Intelligence (AI) and Machine Learning (ML) for the rover moving debris. Adding AI to the construction rover will best open the door to the system being fully autonomous. The AI of the construction rover will consist of separate trained Artificial Neural Network (ANN) for sets of different tasks. An ANN is an AI technique that uses the concept of a biological neuron to weigh parameters for predicting. The first ANN has been trained for the rover's movement based on incoming sensor data producing states for movement. The next ANN is a Convolution Neural Network (CNN) that I am using for object classification, detection, and recognition. A CNN is an ANN, but with the added benefit of a convolutional layer that can take into account the surrounding pixels along with the original pixel into the training of the neural network. The rover's CNN will be trained on images of which objects to carry and which objects to push regarding the AI platform's proposed tasks. The last ANN will be used for reinforcement learning. This ANN is not very different from the first other than it will be trained in a simulation to carry out the important task of path planning. Each ANN gives the AI platform the ability to make its own decisions from what it sees through its camera and reads through its sensors. In the demonstration of the AI construction rover, I hope to show it successfully moving an object using these three AI algorithms as one. I further hope that NASA can use the concepts of these AI algorithms as a tool for the next step of rover programs, and getting humans more opportunities on other worlds.

Objectives and Components

My objectives are to show that the construction rover can complete the tasks of obstacle avoidance, obstacle detection, path planning, and moving an object from one place to another using only its trained autonomy. Briefly speaking these tasks are much easier on Earth, but for a construction rover in space, it must be able to complete its task in a hostile environment and communicate with us humans on Earth from great distances in space. Current rovers also have the added hardship of being moderately controlled by an operator. That's why I'm proposing an autonomous rover using AI to be sent to the rocky bodies that can act in real-time not depending on a team of scientists to catch up to a machine that is 100 million miles away. Now it is easy to use a team to control a rover, but when there are hundreds in space we will need a more solid method.

The rover will be trained with many ANNs that will be used in image recognition, translation of sensor states to movements, and path planning to demonstrate autonomy level 4. Autonomy level 4 behavior is described as high autonomy where that rover is completely autonomous in most situations and conditions with an opening for a human in the loop. NASA Ames's Maria Bualat (2017) Deputy Lead of the Intelligent Robotics Group makes this statement about NASA's relationship with autonomous vehicles saying that we always want the "human in the loop because of experience and powerful cognitive ability". I will be the human component that feeds the ANN real-world and synthetic data made specifically for a rover-like construction robot to complete the tasks above. A better description of this would be, "directed autonomy," to which we can teach the rovers here on Earth then send it anywhere in the Universe, and it can take the human component with it.

Hardware

The prototype construction rover-like robot I will be using in this demonstration is a robot that I have built and named Solar Powered Operation Vehicle Intelligently Autonomous (SPOVIA) Prime shown in Figure 1. I took much inspiration from the 2004 EVA test vehicle shown in Figure 2 that was used to simulate possible futures and technology on a foreign rocky body. I used the tractor scoop idea from their demonstration and added it to SPOVIA prime. SPOVIA prime is a 35-pound solar-powered robot with four strong motors, big wheels, and 30 basic analog sensors consisting of ultrasonic, inferred, ultraviolet, photo, accelerometers, and many more. It also includes the more complex components of lidar, GPS, GSM, stereo camera, and USB color camera. The processor, which is optimal for AI tasks and training like these, is a Jetson TX2 with a Dual-Core NVIDIA Denver 2 64-Bit CPU, Quad-Core ARM® Cortex®-A57 MPCore, and 256 NVIDIA CUDA cores. Also, I will be using the google coral TPU to run the object detection model at speeds up to 40 FPS.

Software

The software I will be using is TensorFlow, PyTorch, and Sci-kit learn for building the ANNs. Pandas, Numpy, and Pymata will be used for supporting those modules in Python 3.7. The synthetic data made will contain states of all of the rover's known action possibilities. The real data collected is from tests of the robot's movements and demonstrations of its actions. The data will be normalized from 0 to 1 among the inputs and outputs and organized into a matrix that contains raw sensor data with the last column being the output state determined to act on. The image data is from the COCO dataset and pictures of rocks and smalls balls I took for the demonstration of classifying and moving an object.

Figure 1 (2004 EVA 'Tractor')

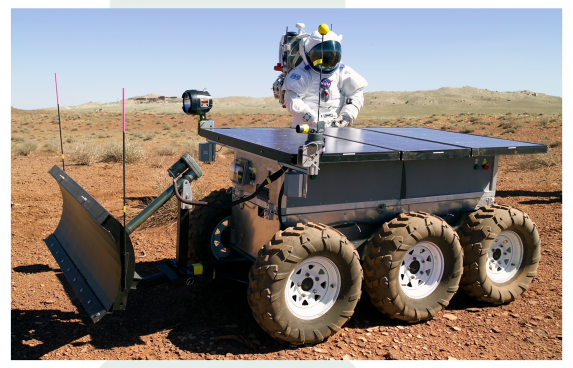
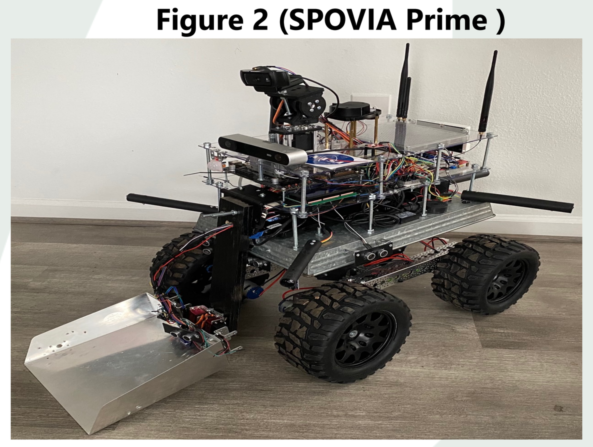


Figure 2 (SPOVIA Prime)



Methods and Preliminary Results

This section shows the algorithms chosen for the construction rover's autonomous decision making and why. Quantitative and qualitative results will be displayed some with graphs and others with pictures of use.

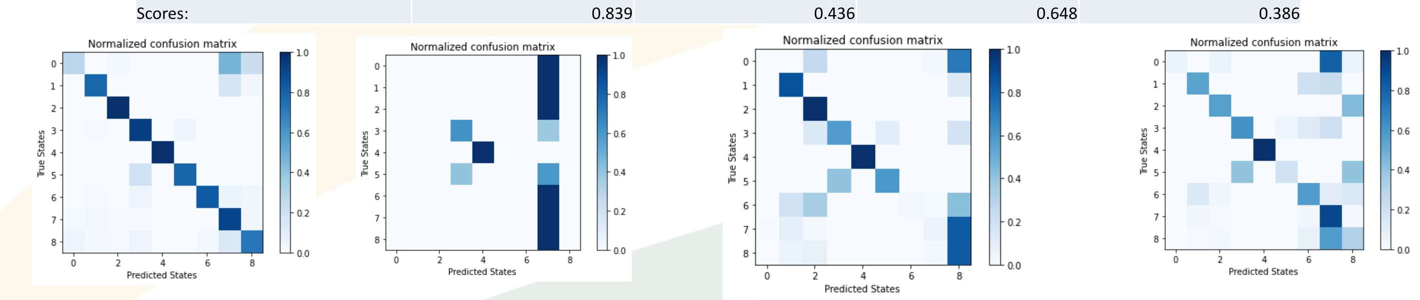
Why Artificial Neural Networks?

As a preliminary effort to choose the best model for the incoming robotic sensor data I compared ANNs with the machine learning techniques Support Vector Machine(SVM), Naive Bias, and Logistic Regression. The resulting model scores and confusion matrices are shown in Figure 3.

Figure 3 (Model Score (80/20 test train split) and normalized confusion matrix of predicted state vs true states.)

ANN SVM Naive Bias Logistic Regression

Names:	ANN	SVM	NB	LR
Scores:	0.839	0.436	0.648	0.386



From the above results, it's clear that an ANN will be the best choice for a predictive tool with the highest model score and best predictive power shown in the first confusion matrix. The next step taken was to make adjustments to the ANN to get the best results and model score for the predicted robot states from the incoming sensor data so that it can best obstacle avoid. I used around 26,000 points of sensor data. Below in Figure 4 is the normalized confusion matrix of the best ANN model with a strong model score of .94.

Figure 4 (Confusion Matrix of Obstacle Avoidance ANN model).

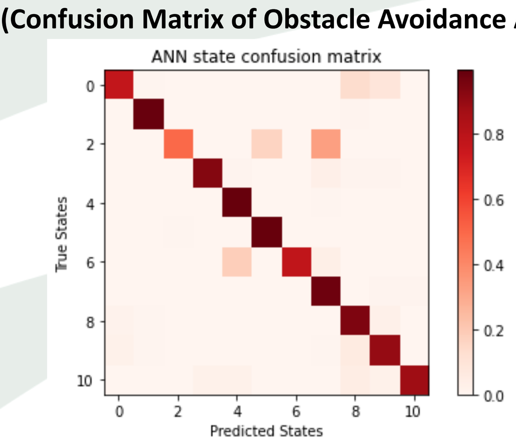
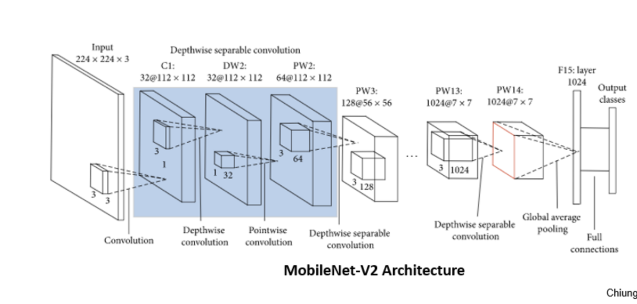


Figure 5(MobileNet SSD v2 CNN structure)



Object Detector using Convolutional Neural Networks

ANNs are not just used for determining sensor states and predicting actions to take. An ANN can also be used to train an object detector, from chosen labeled images, to detect and classify objects. Detecting and classifying gives it the ability to make decisions with the visual information from its surroundings. The ANN I used is similar to the one before except that I added convolutional layers making it a Convolutional Neural Network (CNN). CNNs have the same functions and very similar structures to ANNs, but adding convolutional layers that pool a set of data allows for condensing of important surrounding image features that one would want to be detected. To come up with this result the convolutional layer takes into account the values of close inputs this can be done by simply finding the sum or the average of the incoming inputs aka pooling. CNNs are known to work very well with data in a matrix, like an image, and this is why it is the best choice to be used.

More specifically I am training this object detector using MobileNet SSD v2 using TensorFlow's API. Its structure can be seen in Figure 5 above. I am using TensorFlow to incorporate the new technology of TPUs (Tensor Processing Unit), the google coral, to run my detection algorithm at up to 40 FPS. I have trained it on the COCO dataset plus images of rocks, balls, and other robots. The detection of rocks and balls can be seen in Figure 6. My goal was to train the robot to identify a rock or a ball and initiate a picking up procedure. It uses its scoop and the detected pixel positions to lift the object then travel to the destination of the object and place it down an example of the ball in the scoop is shown in Figure 7.

Figure 6 (Ball & Rock detection)

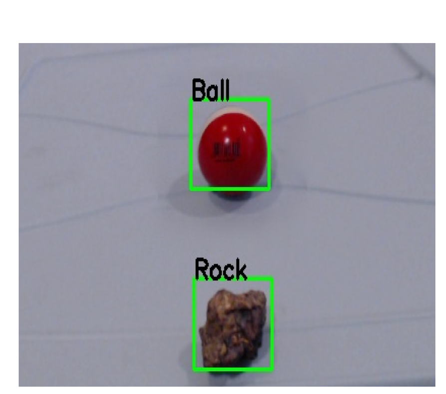


Figure 7(Ball Detection & lift in scoop)

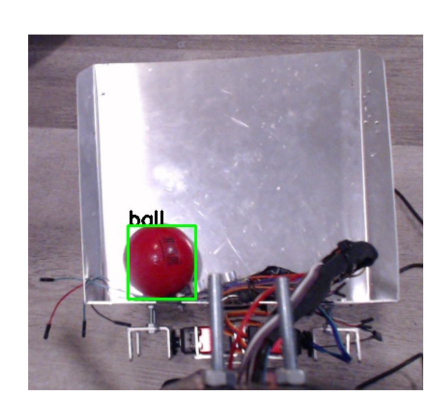
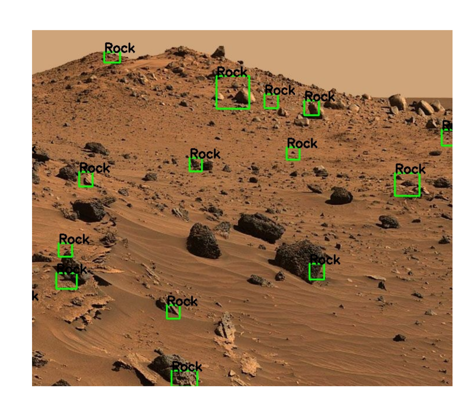


Figure 8(Rock Detection for Mars Surface).



Path Planning

Another important feature that cannot be overlooked is the path planning of the vehicle. This is also a particularly tricky task due to no GPS, roads, and very few land features which creates a very hard problem for an autonomous rover and even a human. The easy solution for a construction rover would be to map the area and create waypoints to travel to and from. I overstepped this problem for SPOVIA Prime using a GPS GUI to plot the points for the rover to travel to, but many obstacles remain especially if it is a large rock in front of the rover that can discourage a straight path. I have attempted to uses the object detection program to detect the rocks in front and possibly avoid them, but seen in Figure 8 this picture of the Martian surface, it is not the most reliable.

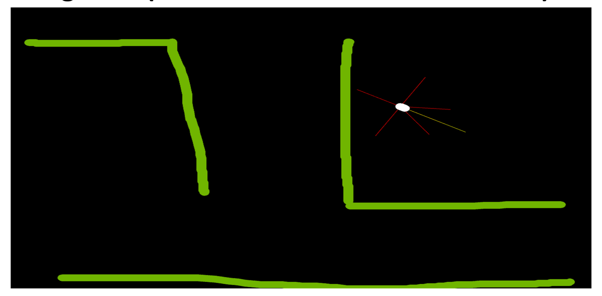
All of my work can be seen at https://www.youtube.com/channel/UCTCYvzO24Ebm68cVsQoy9Eg or MoveOverRover.

Methods and Preliminary Results cont..

Reinforcement learning (RL)

To make the rover go continually straight from one waypoint to another I am using a concept of the algorithm pure pursuit in a reinforcement learning neural network to train the algorithm to do its best to go straight when its sensors don't detect any obstacles. A reinforcement learning neural network is the same as any other ANN, but it is trained in an ongoing simulation with goals, rewards, and point values. Figure 9 is from the 2D simulation of SPOVIA Prime with pure pursuit in mind. The white oval is the rover, the red lines coming outward are the distance sensors, and the yellow line represents a point in front of the rover that is to be followed. The rover gets points based on how close the slope of its initial position and final position is to the slope of the initial position and end of the yellow line. The rover also loses points as the distance sensors (red lines) touch anything. This builds path panning into the fabric of its design allowing for easy, not computationally expensive movement.

Figure 9 (2D Simulation of SPOVIA Prime)



Discussion and Conclusion

Discussion

SPOVIA Prime was very successful at the task of obstacle avoiding using the first ANN. It was a strong rarity that it hit any other detectable objects, although I did have the vehicle travel slower than I would have wanted.
SPOVIA Prime was incredibly successful at detecting the objects it was trained to. From the color camera, it was able to detect rocks, balls, people, and cars (from the COCO dataset) like the examples seen in Figures 6 and 7. But, unfortunately when there were many examples to detect the object detection algorithm failed, seen in Figure 8. The more unfortunate news is that the Martian surface from Figure 8 will be the terrain the construction rover will have to mostly deal with, being that it is meant to be sent to rocky planetary bodies.

A limitation I found is that the simulation avatar had minor differences then SPOVIA Prime, and those differences turned into a problem when moving the algorithm in the simulation to the real world. I believe this is why SPOVIA Prime's movements came out sloppy and jerky, but it did complete the task of obstacle avoidance, object detection, and path planning. I will say that I believe the RL most needs to be updated and changed, but the fault probably most falls on the 2D simulation.

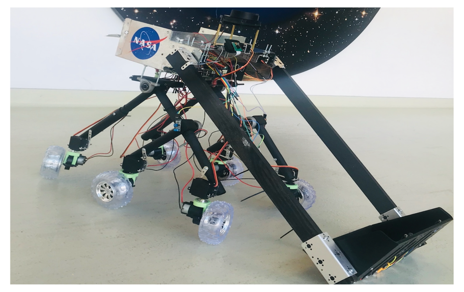
For the main task of picking up a detected object and moving it to another location, it was both a success and a failure. SPOVIA Prime was easily able to detect an object and attempt to pick it. But even when SPOVIA Prime was able to grab the object it was unable to bring it to the chosen waypoint. I ended up settling for success in the robot's ability to detect, lift, and drive away with the detected object still in the scoop.

Conclusion

As a figurative idea, the premise of using AI to make a rover autonomous perform actions is solid, but in the literal, it is much more difficult. Although each algorithm was trained well and was able to perform at its best by itself it was difficult to put together and I rarely got a successful example. The solution to this problem is to better match the algorithm with the physical vehicle. An even bigger problem was deciding which algorithm to carry out and when. I ended up making the obstacle avoidance ANN the priority and when it detected no obstacle the path planning took over followed by the object detector. The solution to this problem would be to make a good task manager for the robotic movements. This was something that I discovered during testing. The takeaways for a more successful autonomous prototype is more training and testing. I believe that in the future this concept will be successful because humanity is curious and will wonder what it is like to live on other celestial bodies, but due to the harsh environments, we will need robotic help to prosper.

Future Work

Figure 10 (SPOVIA Spirit)



I plan to continue this project and demonstrate it again. I will go from a 2D simulation to a 3D, use the object detector CenterNet instead of MobileNet, and create an ANN to be the task manager acting as a decision-maker for the SPOVIA Prime.

After, I plan to use swarm technology for robots to work together. In the object detection section, I mentioned training SPOVIA Prime to see my other robots because I want them to be able to classify each other and work together to get more difficult tasks completed. I have already been working on a partner construction rover seen in Figure 10.

Thank you for your time.

References and Acknowledgements

I would like the thank all the staff of GMU's Physic's department, GMU's The Space Weather Team, and all of the staff at NASA Goddard's Heilophysics lab.

(1) Bualat, 2017, NASA AMES Research center Youtube, https://www.youtube.com/watch?v=P9zmerD04Hk

(2) Rouen et al., (2004), EVA and Mobility Systems Engineering

(3) Saha, 2018, *A comprehensive guide to Convolutional Neural Networks*

(4) Thronson, 2019, *Findings from the Sixth Workshop on Achieving Mars Exploration*